

# pycse - Python3 Computations in Science and Engineering

John Kitchen

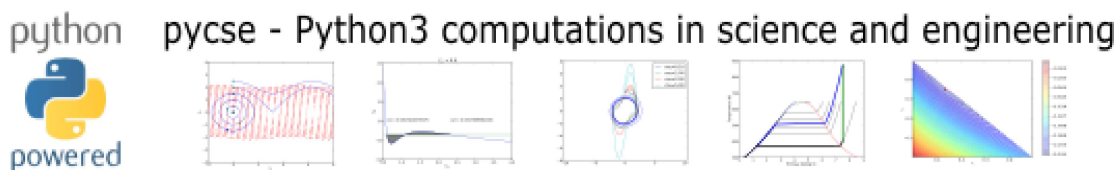
[jkitchen@andrew.cmu.edu](mailto:jkitchen@andrew.cmu.edu)

<https://kitchingroup.cheme.cmu.edu>

Twitter: @johnkitchen

<https://github.com/jkitchen/pycse>

September 22, 2021



## Contents

<b>1</b>	<b>Overview</b>	<b>7</b>
<b>2</b>	<b>Basic python usage</b>	<b>8</b>
2.1	Basic math . . . . .	8
2.2	Advanced mathematical operators . . . . .	8
2.2.1	Exponential and logarithmic functions . . . . .	9
2.3	Creating your own functions . . . . .	9
2.4	Defining functions in python . . . . .	9
2.5	Advanced function creation . . . . .	11
2.6	Lambda Lambda Lambda . . . . .	14
2.6.1	Applications of lambda functions . . . . .	15
2.6.2	Summary . . . . .	16
2.7	Creating arrays in python . . . . .	16
2.8	Functions on arrays of values . . . . .	18
2.9	Some basic data structures in python . . . . .	20
2.9.1	the list . . . . .	20
2.9.2	tuples . . . . .	20
2.9.3	struct . . . . .	20
2.9.4	dictionaries . . . . .	21
2.9.5	Summary . . . . .	21
2.10	Indexing vectors and arrays in Python . . . . .	21
2.10.1	2d arrays . . . . .	22
2.10.2	Using indexing to assign values to rows and columns . . . . .	22
2.10.3	3D arrays . . . . .	23
2.10.4	Summary . . . . .	23

2.11	Controlling the format of printed variables . . . . .	23
2.12	Advanced string formatting . . . . .	25
<b>3</b>	<b>Math</b>	<b>26</b>
3.1	Numeric derivatives by differences . . . . .	26
3.2	Vectorized numeric derivatives . . . . .	28
3.3	2-point vs. 4-point numerical derivatives . . . . .	29
3.4	Derivatives by polynomial fitting . . . . .	30
3.5	Derivatives by fitting a function and taking the analytical derivative . . . . .	32
3.6	Derivatives by FFT . . . . .	34
3.7	A novel way to numerically estimate the derivative of a function - complex-step derivative approximation . . . . .	35
3.8	Vectorized piecewise functions . . . . .	36
3.9	Smooth transitions between discontinuous functions . . . . .	40
3.9.1	Summary . . . . .	44
3.10	Smooth transitions between two constants . . . . .	44
3.11	On the quad or trapz'd in ChemE heaven . . . . .	45
3.11.1	Numerical data integration . . . . .	46
3.11.2	Combining numerical data with quad . . . . .	47
3.11.3	Summary . . . . .	48
3.12	Polynomials in python . . . . .	48
3.12.1	Summary . . . . .	49
3.13	Wilkinson's polynomial . . . . .	49
3.14	The trapezoidal method of integration . . . . .	52
3.15	Numerical Simpsons rule . . . . .	53
3.16	Integrating functions in python . . . . .	54
3.16.1	double integrals . . . . .	54
3.16.2	Summary . . . . .	55
3.17	Integrating equations in python . . . . .	55
3.18	Function integration by the Romberg method . . . . .	56
3.19	Symbolic math in python . . . . .	56
3.19.1	Solve the quadratic equation . . . . .	56
3.19.2	differentiation . . . . .	56
3.19.3	integration . . . . .	56
3.19.4	Analytically solve a simple ODE . . . . .	57
3.20	Is your ice cream float bigger than mine . . . . .	57
<b>4</b>	<b>Linear algebra</b>	<b>58</b>
4.1	Potential gotchas in linear algebra in numpy . . . . .	58
4.2	Solving linear equations . . . . .	60
4.3	Rules for transposition . . . . .	61
4.3.1	The transpose in Python . . . . .	61
4.3.2	Rule 1 . . . . .	61
4.3.3	Rule 2 . . . . .	61
4.3.4	Rule 3 . . . . .	62
4.3.5	Rule 4 . . . . .	62
4.3.6	Summary . . . . .	62
4.4	Sums products and linear algebra notation - avoiding loops where possible . . . . .	62

4.4.1	Old-fashioned way with a loop . . . . .	62
4.4.2	The numpy approach . . . . .	63
4.4.3	Matrix algebra approach. . . . .	63
4.4.4	Another example . . . . .	63
4.4.5	Last example . . . . .	64
4.4.6	Summary . . . . .	64
4.5	Determining linear independence of a set of vectors . . . . .	64
4.5.1	another example . . . . .	65
4.5.2	Near deficient rank . . . . .	66
4.5.3	Application to independent chemical reactions. . . . .	66
4.6	Reduced row echelon form . . . . .	67
4.7	Computing determinants from matrix decompositions . . . . .	68
4.8	Calling lapack directly from scipy . . . . .	68
<b>5</b>	<b>Nonlinear algebra</b>	<b>70</b>
5.1	Know your tolerance . . . . .	70
5.2	Solving integral equations with fsolve . . . . .	71
5.2.1	Summary notes . . . . .	73
5.3	Method of continuity for nonlinear equation solving . . . . .	73
5.4	Method of continuity for solving nonlinear equations - Part II . . . . .	76
5.5	Counting roots . . . . .	78
5.5.1	Use roots for this polynomial . . . . .	78
5.5.2	method 1 . . . . .	79
5.5.3	Method 2 . . . . .	80
5.6	Finding the nth root of a periodic function . . . . .	80
5.7	Coupled nonlinear equations . . . . .	82
<b>6</b>	<b>Statistics</b>	<b>83</b>
6.1	Introduction to statistical data analysis . . . . .	83
6.2	Basic statistics . . . . .	84
6.3	Confidence interval on an average . . . . .	84
6.4	Are averages different . . . . .	85
6.4.1	The hypothesis . . . . .	85
6.4.2	Compute the t-score for our data . . . . .	86
6.4.3	Interpretation . . . . .	86
6.5	Model selection . . . . .	87
6.6	Numerical propagation of errors . . . . .	94
6.6.1	Addition and subtraction . . . . .	95
6.6.2	Multiplication . . . . .	95
6.6.3	Division . . . . .	95
6.6.4	exponents . . . . .	95
6.6.5	the chain rule in error propagation . . . . .	96
6.6.6	Summary . . . . .	96
6.7	Another approach to error propagation . . . . .	96
6.7.1	Summary . . . . .	99
6.8	Random thoughts . . . . .	99
6.8.1	Summary . . . . .	102

<b>7</b>	<b>Data analysis</b>	<b>102</b>
7.1	Fit a line to numerical data . . . . .	102
7.2	Linear least squares fitting with linear algebra . . . . .	103
7.3	Linear regression with confidence intervals (updated) . . . . .	104
7.4	Linear regression with confidence intervals. . . . .	105
7.5	Nonlinear curve fitting . . . . .	106
7.6	Nonlinear curve fitting by direct least squares minimization . . . . .	108
7.7	Parameter estimation by directly minimizing summed squared errors . . . . .	108
7.8	Nonlinear curve fitting with parameter confidence intervals . . . . .	112
7.9	Nonlinear curve fitting with confidence intervals . . . . .	114
7.10	Graphical methods to help get initial guesses for multivariate nonlinear regression . . . . .	115
7.11	Fitting a numerical ODE solution to data . . . . .	119
7.12	Reading in delimited text files . . . . .	120
<b>8</b>	<b>Interpolation</b>	<b>121</b>
8.1	Better interpolate than never . . . . .	121
8.1.1	Estimate the value of $f$ at $t=2$ . . . . .	121
8.1.2	improved interpolation? . . . . .	122
8.1.3	The inverse question . . . . .	123
8.1.4	A harder problem . . . . .	124
8.1.5	Discussion . . . . .	125
8.2	Interpolation of data . . . . .	126
8.3	Interpolation with splines . . . . .	127
<b>9</b>	<b>Optimization</b>	<b>127</b>
9.1	Constrained optimization . . . . .	127
9.2	Finding the maximum power of a photovoltaic device. . . . .	129
9.3	Using Lagrange multipliers in optimization . . . . .	131
9.3.1	Construct the Lagrange multiplier augmented function . . . . .	132
9.3.2	Finding the partial derivatives . . . . .	133
9.3.3	Now we solve for the zeros in the partial derivatives . . . . .	133
9.3.4	Summary . . . . .	133
9.4	Linear programming example with inequality constraints . . . . .	133
9.5	Find the minimum distance from a point to a curve. . . . .	135
<b>10</b>	<b>Differential equations</b>	<b>136</b>
10.1	Ordinary differential equations . . . . .	137
10.1.1	Numerical solution to a simple ode . . . . .	137
10.1.2	Plotting ODE solutions in cylindrical coordinates . . . . .	138
10.1.3	ODEs with discontinuous forcing functions . . . . .	140
10.1.4	Simulating the events feature of Matlab's ode solvers . . . . .	141
10.1.5	Mimicking ode events in python . . . . .	143
10.1.6	Solving an ode for a specific solution value . . . . .	146
10.1.7	A simple first order ode evaluated at specific points . . . . .	149
10.1.8	Stopping the integration of an ODE at some condition . . . . .	149
10.1.9	Finding minima and maxima in ODE solutions with events . . . . .	150
10.1.10	Error tolerance in numerical solutions to ODEs . . . . .	151
10.1.11	Solving parameterized ODEs over and over conveniently . . . . .	154

10.1.12	Yet another way to parameterize an ODE . . . . .	155
10.1.13	Another way to parameterize an ODE - nested function . . . . .	157
10.1.14	Solving a second order ode . . . . .	158
10.1.15	Solving Bessel's Equation numerically . . . . .	161
10.1.16	Phase portraits of a system of ODEs . . . . .	162
10.1.17	Linear algebra approaches to solving systems of constant coefficient ODEs . .	165
10.2	Delay Differential Equations . . . . .	167
10.3	Differential algebraic systems of equations . . . . .	167
10.4	Boundary value equations . . . . .	167
10.4.1	Plane Poiseuille flow - BVP solve by shooting method . . . . .	167
10.4.2	Plane poiseuille flow solved by finite difference . . . . .	172
10.4.3	Boundary value problem in heat conduction . . . . .	175
10.4.4	BVP in pycse . . . . .	177
10.4.5	A nonlinear BVP . . . . .	179
10.4.6	Another look at nonlinear BVPs . . . . .	182
10.4.7	Solving the Blasius equation . . . . .	184
10.5	Partial differential equations . . . . .	186
10.5.1	Modeling a transient plug flow reactor . . . . .	186
10.5.2	Transient heat conduction - partial differential equations . . . . .	189
10.5.3	Transient diffusion - partial differential equations . . . . .	192
<b>11</b>	<b>Plotting</b>	<b>195</b>
11.1	Plot customizations - Modifying line, text and figure properties . . . . .	195
11.1.1	setting all the text properties in a figure. . . . .	198
11.2	Plotting two datasets with very different scales . . . . .	200
11.2.1	Make two plots! . . . . .	200
11.2.2	Scaling the results . . . . .	202
11.2.3	Double-y axis plot . . . . .	203
11.2.4	Subplots . . . . .	204
11.3	Customizing plots after the fact . . . . .	205
11.4	Fancy, built-in colors in Python . . . . .	209
11.5	Picasso's short lived blue period with Python . . . . .	209
11.6	Interactive plotting . . . . .	211
11.6.1	Basic mouse clicks . . . . .	211
11.7	key events not working on Mac/org-mode . . . . .	214
11.7.1	Mouse movement . . . . .	215
11.7.2	key press events . . . . .	216
11.7.3	Picking lines . . . . .	216
11.7.4	Picking data points . . . . .	217
11.8	Peak annotation in matplotlib . . . . .	218
<b>12</b>	<b>Programming</b>	<b>220</b>
12.1	Some of this, sum of that . . . . .	220
12.1.1	Nested lists . . . . .	221
12.2	Sorting in python . . . . .	221
12.3	Unique entries in a vector . . . . .	222
12.4	Lather, rinse and repeat . . . . .	223
12.4.1	Conclusions . . . . .	224

12.5	Brief intro to regular expressions . . . . .	224
12.6	Working with lists . . . . .	225
12.7	Making word files in python . . . . .	226
12.8	Interacting with Excel in python . . . . .	227
12.8.1	Writing Excel workbooks . . . . .	228
12.8.2	Updating an existing Excel workbook . . . . .	228
12.8.3	Summary . . . . .	229
12.9	Using Excel in Python . . . . .	229
12.10	Running Aspen via Python . . . . .	230
12.11	Using an external solver with Aspen . . . . .	232
12.12	Redirecting the print function . . . . .	233
12.13	Getting a dictionary of counts . . . . .	235
12.14	About your python . . . . .	236
12.15	Automatic, temporary directory changing . . . . .	236
<b>13</b>	<b>Miscellaneous</b>	<b>238</b>
13.1	Mail merge with python . . . . .	238
<b>14</b>	<b>Worked examples</b>	<b>239</b>
14.1	Peak finding in Raman spectroscopy . . . . .	239
14.1.1	Summary notes . . . . .	243
14.2	Curve fitting to get overlapping peak areas . . . . .	243
14.2.1	Notable differences from Matlab . . . . .	248
14.3	Estimating the boiling point of water . . . . .	248
14.3.1	Summary . . . . .	250
14.4	Gibbs energy minimization and the NIST webbook . . . . .	251
14.4.1	Compute mole fractions and partial pressures . . . . .	252
14.4.2	Computing equilibrium constants . . . . .	252
14.5	Finding equilibrium composition by direct minimization of Gibbs free energy on mole numbers . . . . .	253
14.5.1	The Gibbs energy of a mixture . . . . .	253
14.5.2	Linear equality constraints for atomic mass conservation . . . . .	253
14.5.3	Equilibrium constant based on mole numbers . . . . .	254
14.5.4	Summary . . . . .	255
14.6	The Gibbs free energy of a reacting mixture and the equilibrium composition . . . . .	255
14.6.1	Summary . . . . .	259
14.7	Water gas shift equilibria via the NIST Webbook . . . . .	259
14.7.1	hydrogen . . . . .	259
14.7.2	H <sub>2</sub> O . . . . .	260
14.7.3	CO . . . . .	260
14.7.4	CO <sub>2</sub> . . . . .	260
14.7.5	Standard state heat of reaction . . . . .	261
14.7.6	Non-standard state $\Delta H$ and $\Delta G$ . . . . .	261
14.7.7	Plot how the $\Delta G$ varies with temperature . . . . .	261
14.7.8	Equilibrium constant calculation . . . . .	262
14.7.9	Equilibrium yield of WGS . . . . .	263
14.7.10	Compute gas phase pressures of each species . . . . .	264
14.7.11	Compare the equilibrium constants . . . . .	264

14.7.12	Summary . . . . .	264
14.8	Constrained minimization to find equilibrium compositions . . . . .	264
14.8.1	summary . . . . .	267
14.9	Using constrained optimization to find the amount of each phase present . . . . .	268
14.10	Conservation of mass in chemical reactions . . . . .	270
14.11	Numerically calculating an effectiveness factor for a porous catalyst bead . . . . .	270
14.12	Computing a pipe diameter . . . . .	273
14.13	Reading parameter database text files in python . . . . .	274
14.14	Calculating a bubble point pressure of a mixture . . . . .	277
14.15	The equal area method for the van der Waals equation . . . . .	278
14.15.1	Compute areas . . . . .	281
14.16	Time dependent concentration in a first order reversible reaction in a batch reactor .	283
14.17	Finding equilibrium conversion . . . . .	284
14.18	Integrating a batch reactor design equation . . . . .	285
14.19	Uncertainty in an integral equation . . . . .	285
14.20	Integrating the batch reactor mole balance . . . . .	286
14.21	Plug flow reactor with a pressure drop . . . . .	287
14.22	Solving CSTR design equations . . . . .	288
14.23	Meet the steam tables . . . . .	289
14.23.1	Starting point in the Rankine cycle in condenser. . . . .	289
14.23.2	Isentropic compression of liquid to point 2 . . . . .	289
14.23.3	Isobaric heating to T3 in boiler where we make steam . . . . .	290
14.23.4	Isentropic expansion through turbine to point 4 . . . . .	290
14.23.5	To get from point 4 to point 1 . . . . .	290
14.23.6	Efficiency . . . . .	290
14.23.7	Entropy-temperature chart . . . . .	290
14.23.8	Summary . . . . .	292
14.24	What region is a point in . . . . .	292
<b>15</b>	<b>Units</b>	<b>298</b>
15.1	Using units in python . . . . .	298
15.1.1	scimath . . . . .	299
15.2	Handling units with the quantities module . . . . .	300
15.3	Units in ODEs . . . . .	303
15.4	Handling units with dimensionless equations . . . . .	307
<b>16</b>	<b>GNU Free Documentation License</b>	<b>309</b>
<b>17</b>	<b>Additional References</b>	<b>318</b>
<b>Index</b>		<b>319</b>

## 1 Overview

This is a collection of examples of using python in the kinds of scientific and engineering computations I have used in classes and research. They are organized by topics.

I recommend the Continuum IO Anaconda python distribution (<https://www.continuum.io>). This distribution is free for academic use, and cheap otherwise. It is pretty complete in terms of